A Comparative Study of Generators of Synthetic Self-Similar Teletraffic*

H.-D. J. Jeong†, D. McNickle‡ and K. Pawlikowski†

Department of †Computer Science and ‡Management
University of Canterbury
Christchurch, New Zealand
e-mail: {joshua, krys@cosc, d.mcnickle@mang}.canterbury.ac.nz
Ph.: +(64) 3 364 2362, Fax.: +(64) 3 364 2569

Abstract

It is generally accepted that *self-similar* (or *fractal*) processes may provide better models for teletraffic in modern telecommunication networks than Poisson processes. If this is not taken into account, it can lead to inaccurate conclusions about performance of telecommunication networks. Thus, an important requirement for conducting simulation studies of telecommunication networks is the ability to generate long synthetic stochastic self-similar sequences.

Three generators of pseudo-random self-similar sequences, based on the FFT [19], RMD [12] and SRA method [5], [10], are compared and analysed in this paper. Properties of these generators were experimentally studied in the sense of their statistical accuracy and times required to produce sequences of a given (long) length. While all three generators show similar levels of accuracy of the output data (in the sense of relative accuracy of the Hurst parameter), the RMD- and SRA-based generators appear to be much faster than the generator based on FFT. Our results also show that a robust method for comparative studies of self-similarity in pseudo-random sequences is needed.

1 Introduction

The search for accurate mathematical models of data streams in modern telecommunication networks has attracted a considerable amount of interest in the last few years. The reason is that several recent teletraffic studies of local and wide area networks, including the world wide web, have shown that commonly used teletraffic models, based on Poisson or related processes, are not able to capture the self-similar (or fractal) nature of teletraffic [13], [14], [20], [25], especially when they are engaged in such sophisticated services as variable-bit-rate (VBR) video transmission [6], [11], [24]. The properties of teletraffic in such scenarios are very different from both the properties of conventional models of telephone traffic and the traditional models of data traffic generated by computers.

The use of traditional models of teletraffic can result in overly optimistic estimates of performance of telecommunication networks, insufficient allocation of communication and data processing resources, and difficulties in ensuring the quality of service expected by network users [1], [17], [20]. On the other hand, if the strongly correlated character of teletraffic is explicitly taken into account, this can also lead to more efficient traffic control mechanisms.

*Technical Report TR-COSC 03/98

Several methods for generating pseudo-random self-similar sequences have been proposed. They include methods based on fast fractional Gaussian noise [15], fractional ARIMA processes [9], the $M/G/\infty$ queue model [11], [13], autoregressive processes [3], [8], spatial renewal processes [26], etc. Some of them generate asymptotically self-similar sequences and require large amounts of CPU time. For example, Hosking's method [9], based on the F-ARIMA$(0, d, 0)$ process, needs many hours to produce a self-similar sequence with 131,072 (2^{17}) numbers on a Sun SPARCstation 4 [13]. It requires $O(n^2)$ computations to generate n numbers. Even though exact methods of generation of self-similar sequences exist (for example: [15]), they are only fast enough for short sequences. They are usually inappropriate for generating long sequences because they require multiple passes along generated sequences. To overcome this, approximate methods for generation of self-similar sequences in simulation studies of telecommunication networks have also been proposed [12], [19].

Our comparative evaluation of three methods proposed for generating self-similar sequences concentrates on two aspects: (i) how accurately self-similar processes can be generated, and (ii) how fast the methods generate long self-similar sequences. We consider three methods: (i) a method based on the *fast Fourier transform (FFT)* algorithm and implemented by Paxson[19]; (ii) a method based on the *random midpoint displacement (RMD)* algorithm and implemented by Lau, Erramilli, Wang and Willinger [12]; and (iii) a method based on the *successive random addition (SRA)* algorithm, proposed by Saupe, D. [5] and implemented by Jeong, McNickle and Pawlikowski [10].

A summary of the basic properties of self-similar processes is given in section 2. In section 3 the three generators of pseudo-random self-similar sequences are described. Numerical results of comparative analysis of sequences generated by these generators are discussed in section 4.

2 Self-Similar Processes and Their Properties

Basic definitions of self-similar processes are as follows:

A continuous-time stochastic process $\{X_t\}$ is strongly *self-similar* with a self-similarity parameter $H (0 < H < 1)$, know as the Hurst parameter, if for any positive stretching factor c, the rescaled process with time scale $ct, c^{-H}X_{ct}$, is equal in distribution to the original process $\{X_t\}$ [2]. This means that, for any sequence of time points $t_1, t_2, \ldots, t_n$, and for all $c > 0$, $\{c^{-H}X_{ct_1}, c^{-H}X_{ct_2}, \ldots, c^{-H}X_{ct_n}\}$ has the same distribution as $\{X_{t_1}, X_{t_2}, \ldots, X_{t_n}\}$.

In discrete-time case, let $\{X_k\} = \{X_k : k = 0, 1, 2, \ldots\}$ be a (discrete-time) stationary process with mean μ, variance σ^2, and autocorrelation function (ACF) $\{\rho_k\}$, for $k = 0, 1, 2, \ldots$, and let $\{X_k^{(m)}\}_{k=1}^{\infty} = \{X_1^{(m)}, X_2^{(m)}, \ldots\}$, $m = 1, 2, 3, \ldots$, be a sequence of batch means, i.e., $X_k^{(m)} = (X_{km-m+1} + \ldots + X_{km})/m, k \geq 1$.

The process $\{X_k\}$ with $\rho_k \to k^{-\beta}$, as $k \to \infty, 0 < \beta < 1$, is called *exactly self-similar* with $H = 1 - (\beta/2)$, if $\rho_k^{(m)} = \rho_k$, for any $m = 1, 2, 3, \ldots$. In other words, the process $\{X_k\}$ and the averaged processes $\{X_k^{(m)}\}$, $m \geq 1$, have identical correlation structure.

The process $\{X_k\}$ is *asymptotically self-similar* with $H = 1 - (\beta/2)$, if $\rho_k^{(m)} \to \rho_k$, as $m \to \infty$.

The most frequently studied models of self-similar traffic belong either to the class of fractional autoregressive integrated moving-average (F-ARIMA) processes or to the class of fractional Gaussian noise processes; see [9], [13], [19]. F-ARIMA(p, d, q) processes were introduced by Hosking [9] who showed that they are asymptotically self-similar with Hurst parameter $H = d + \frac{1}{2}$, as long as $0 < d < \frac{1}{2}$. In addition, the incremental process $\{Y_k\} = \{X_k - X_{k-1}\}, k \geq 0$, is called the *fractional Gaussian noise* (FGN) process, where $\{X_k\}$ designates a fractional Brownian motion (FBM) random process. This process is a (discrete-time) stationary Gaussian process with mean μ, variance σ^2 and

$\{\rho_k\} = \{\frac{1}{2}(|k+1|^{2H} - 2|k|^{2H} + |k-1|^{2H})\}, \quad k > 0$. A FBM process, which is the sum of FGN increments, is characterised by three properties [16]: (i) it is a continuous zero-mean Gaussian process $\{X_t\} = \{X_s : s \geq 0 \text{ and } \quad 0 < H < 1\}$ with ACF given by $\rho_{s,t} = \frac{1}{2}(s^{2H} + t^{2H} - |s-t|^{2H})$ where s is time lag and t is time; (ii) its increments $\{X_t - X_{t-1}\}$ form a stationary random process; (iii) it is self-similar with Hurst parameter H, that is, for all $c > 0$, $\{X_{ct}\} = \{c^H X_t\}$, in the sense that, if time is changed by the ratio c, then $\{X_t\}$ is changed by c^H.

Main properties of self-similar processes include ([2], [4], [13]):

- *Slowly decaying variance.* The variance of the sample mean decreases more slowly than the reciprocal of the sample size, that is, $Var[\{X_k^{(m)}\}] \to c_1 m^{-\beta_1}$ as $m \to \infty$, where c_1 is a constant and $0 < \beta_1 < 1$.
- *Long-range dependence.* A process $\{X_k\}$ is called a stationary process with *long-range dependence (LRD)* if its ACF $\{\rho_k\}$ is non-summable, that is, $\sum_{k=0}^{\infty} \rho_k = \infty$. The speed of decay of autocorrelations is more like hyperbolic than exponential.
- *Hurst effect.* Self-similarity manifests itself by a straight line of slope β_2 on a log-log plot of the R/S statistic. For a given set of numbers $\{X_1, X_2, \ldots, X_n\}$ with sample mean $\hat{\mu} = E\{X_i\}$ and sample variance $S^2(n) = E\{(X_i - \hat{\mu})^2\}$, Hurst parameter H is presented by the *rescaled adjusted range* $\frac{R(n)}{S(n)}$ (or R/S *statistic*) where $R(n) = \max\{\sum_{i=1}^{k}(X_i - \hat{\mu}), 1 \leq k \leq n\} - \min\{\sum_{i=1}^{k}(X_i - \hat{\mu}), 1 \leq k \leq n\}$ and S is estimated by $S(n) = \sqrt{E\{(X_i - \hat{\mu})^2\}}$. Hurst found empirically that for many time series observed in nature the expected value of $\frac{R(n)}{S(n)}$ asymptotically satisfies the power law relation, i.e., $E[\frac{R(n)}{S(n)}] \to c_2 n^H$ as $n \to \infty$ with $0.5 < H < 1$ and c_2 is a finite positive constant [2].

In simulation of telecommunication networks, given a sequence of the approximate FBM process $\{X_t\}$, we can obtain a self-similar cumulative arrival process $\{Y_t\}$ [12], [18]: $\{Y_t\} = Mt + \sqrt{AM}\{X_t\}, \quad t \in (-\infty, +\infty)$ where M is the mean input rate and A is the peakedness factor, defined as the ratio of variance to the mean, $M > 0, A > 0$. The Gaussian incremental process $\{\tilde{Y}_t\}$ from time t to time $t+1$ is given as: $\{\tilde{Y}_t\} = M + \sqrt{AM}[\{X_{t+1}\} - \{X_t\}]$.

3 Three Methods

The FFT- and RMD-based methods were suggested as being sufficiently fast for practical applications in generation of simulation input data [12], [19]. In this paper, we have reported properties of these two methods and compare them with SRA, one of recently proposed alternative methods for generating pseudo-random self-similar sequences [10]. These methods can be characterised as follows:

3.1 FFT method

This method generates approximate self-similar sequences based on the Fast Fourier Transform and a process known as the Fractional Gaussian Noise (FGN) process, (Figure 1.) Its main difficulty is connected with calculating the power spectrum, which involves an infinite summation. Paxson has solved this problem by applying a special approximation.

Figure 1 shows how the FFT method generates self-similar sequences. Briefly, it is based on (i) calculation of the power spectrum using the periodogram (the power spectrum at a given frequency

represents an independent exponential random variable); (ii) construction of complex numbers which are governed by the normal distribution; (iii) execution of the inverse FFT. An overview of the FFT method to generate sequences given below, follows. For a more detailed reference, see [19].

This leads to the following algorithm:

Step.1 Generate a sequence of values $\{f_1, \ldots, f_{\frac{n}{2}}\}$, where $f_i = \hat{f}(\frac{2\pi i}{n}, H)$, corresponding to the power spectrum of a FGN process for frequencies from $\frac{2\pi}{n}$ to π, $1/2 < H < 1$.

For a FGN process, the power spectrum $f(\lambda, H)$ is defined as

$$f(\lambda, H) = \mathcal{A}(\lambda, H)[|\lambda|^{-2H-1} + \mathcal{B}(\lambda, H)], \tag{1}$$

for $0 < H < 1$ and $-\pi \leq \lambda \leq \pi$, where

$$\begin{aligned} \mathcal{A}(\lambda, H) &= 2sin(\pi H)\Gamma(2H+1)(1-cos\lambda), \\ \mathcal{B}(\lambda, H) &= \sum_{i=1}^{\infty}[(2\pi i + \lambda)^{-2H-1} + (2\pi i - \lambda)^{-2H-1}]. \end{aligned} \tag{2}$$

As mentioned the infinite summation in Equation (1) for $\mathcal{B}(\lambda, H)$ is the main difficulty in computing the power spectrum. He [19] proposed to use the approximation given by Equation (3) instead of Equation (2):

$$\mathcal{B}(\lambda, H) \approx a_1^d + b_1^d + a_2^d + b_2^d + a_3^d + b_3^d + \frac{a_3^{d'} + b_3^{d'} + a_4^{d'} + b_4^{d'}}{8H\pi} \tag{3}$$

where $d = -2H - 1, d' = -2H, a_i = 2i\pi + \lambda, b_i = 2i\pi - \lambda$.

Step.2 Adjust the sequence of values $\{f_1, \ldots, f_{\frac{n}{2}}\}$ for estimating power spectrum using periodogram.

Step.3 Generate $\{Z_1, \ldots, Z_{\frac{n}{2}}\}$, a sequence of complex values such that $|Z_i| = \sqrt{\hat{f}_i}$ and the phase of Z_i is uniformly distributed between 0 and 2π.

Step.4 Construct $\{Z_0', \ldots, Z_{n-1}'\}$, an *expanded* version of $\{Z_1, \ldots, Z_{\frac{n}{2}}\}$:

$$Z_i' = \begin{cases} 0, & \text{if } i = 0, \\ Z_i, & \text{if } 0 < i \leq \frac{n}{2}, \text{and} \\ \overline{Z_{n-i}}, & \text{if } \frac{n}{2} < i < n. \end{cases} \tag{4}$$

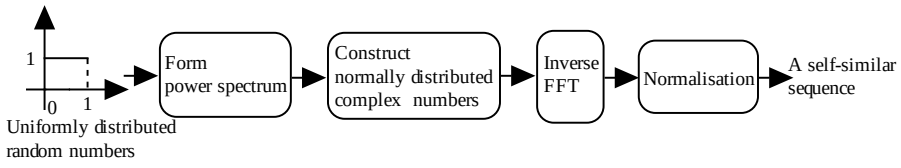

Figure 1: FFT method

where $\overline{Z_{n-i}}$ denotes the complex conjugate of Z_{n-i}. $\{Z_i^{'}\}$ retains the power spectrum used in constructing $\{Z_i\}$, but because it is symmetric about $Z_{\frac{n}{2}}^{'}$, it now corresponds to the FFT of a real-valued signal.

Step.5 Calculate inverse FFT $\{Z_i^{'}\}$ to obtain the approximate FGN sequence $\{X_i\}$.

3.2 RMD method

The basic concept of the *random midpoint displacement (RMD)* algorithm is to extend the generated sequence recursively, by adding new values at the midpoints from the values at the endpoints.

Figure 2 outlines how the RMD algorithm works. Figure 3 illustrates the first three steps of the method, leading to generation of the sequence $(d_{3,1}, d_{3,2}, d_{3,3}, d_{3,4})$. The reason for subdividing the interval between 0 and 1 is to construct the Gaussian increments of X. Adding offsets to midpoints makes the marginal distribution of the final result normal. For more detailed discussions of the RMD method, see [12], [21].

Step.1 If the process $X(t)$ is to be computed for time instance t between 0 and 1, then start out by setting $X(0) = 0$ and selecting $X(1)$ as a pseudo-random number from a Gaussian distribution with mean 0 and variance $Var[X(1)] = \sigma_0^2$. Then $Var[X(1) - X(0)] = \sigma_0^2$.

Step.2 Next, $X(\frac{1}{2})$ is constructed as the average of $X(0)$ and $X(1)$, that is, $X(\frac{1}{2}) = \frac{1}{2}(X(0)+X(1))+d_1$. The offset d_1 is a Gaussian random number (GRN), which should be multiplied by a scaling factor $\frac{1}{2}$, with mean 0 and variance S_1^2 of d_1. Compare the visualisation of this step and the next one in Figure 3. For $Var[X(t_2) - X(t_1)] = |t_2 - t_1|^{2H}\sigma_0^2$ to be true, for $0 \le t_1 \le t_2 \le 1$, it must be required that $Var[X(\frac{1}{2}) - X(0)] = \frac{1}{4}Var[X(1) - X(0)] + S_1^2 = (\frac{1}{2})^{2H}\sigma_0^2$. Thus $S_1^2 = (\frac{1}{2^1})^{2H}(1 - 2^{2H-2})\sigma_0^2$.

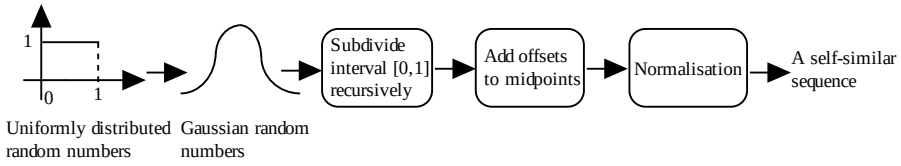

Figure 2: RMD method

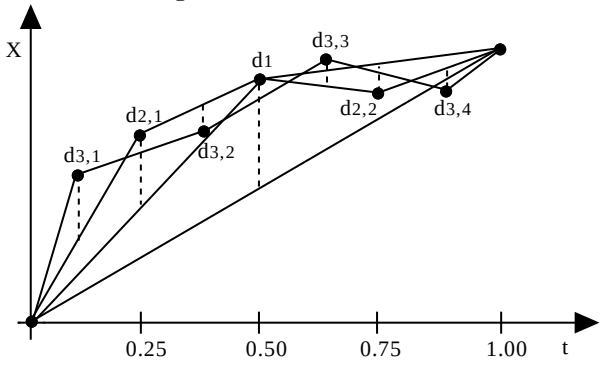

Figure 3: The first three steps in the RMD method

Step.3 Reduce the scaling factor by $\sqrt{2}$, that is, now assume $\frac{1}{\sqrt{8}}$, and divide the two intervals from 0 and $\frac{1}{2}$ and from $\frac{1}{2}$ to 1 again. $X(\frac{1}{4})$ is set as the average $\frac{1}{2}(X(0)+X(\frac{1}{2}))$ plus an offset $d_{2,1}$, which is a GRN multiplied by the current scaling factor $\frac{1}{\sqrt{8}}$. The corresponding formula holds for $X(\frac{3}{4})$, that is, $X(\frac{3}{4}) = \frac{1}{2}(X(\frac{1}{2})+X(1))+d_{2,2}$ where $d_{2,2}$ is a random offset computed as before. So the variance S_2^2 of $d_{2,*}$ must be chosen such that $Var[X(\frac{1}{4})-X(0)] = \frac{1}{4}Var[X(\frac{1}{2})-X(0)]+S_2^2 = (\frac{1}{2^2})^{2H}\sigma_0^2$. Thus $S_2^2 = (\frac{1}{2^2})^{2H}(1-2^{2H-2})\sigma_0^2$.

Step.4 The fourth step proceeds in the same manner: reduce the scaling factor by $\sqrt{2}$, that is, do $\frac{1}{\sqrt{16}}$. Then set

$$X(\frac{1}{8}) = \frac{1}{2}(X(0)+X(\frac{1}{4}))+d_{3,1}, X(\frac{3}{8}) = \frac{1}{2}(X(\frac{1}{4})+X(\frac{1}{2}))+d_{3,2}$$
$$X(\frac{5}{8}) = \frac{1}{2}(X(\frac{1}{2})+X(\frac{3}{4}))+d_{3,3}, X(\frac{7}{8}) = \frac{1}{2}(X(\frac{3}{4})+X(1))+d_{3,4}.$$

In each formula, $d_{3,*}$ is computed as a different GRN multiplied by the current scaling factor $\frac{1}{\sqrt{16}}$. The following step computes $X(t)$ at $t = \frac{1}{16}, \frac{3}{16}, \ldots, \frac{15}{16}$ using a scaling factor again reduced by $\sqrt{2}$, and continues as indicated above. So the variance S_3^2 of $d_{3,*}$ must be chosen such that $Var(X(\frac{1}{8})-X(0)) = \frac{1}{4}Var(X(\frac{1}{4})-X(0))+S_3^2 = (\frac{1}{2^3})^{2H}\sigma_0^2$, that is, $S_3^2 = (\frac{1}{2^3})^{2H}(1-2^{2H-2})\sigma_0^2$. The variance S_n^2 of $d_{n,*}$, therefore, yields $(\frac{1}{2^n})^{2H}(1-2^{2H-2})\sigma_0^2$.

3.3 SRA method

Another alternative method for the direct generation of FBM process is based on the *successive random addition* (SRA) algorithm [5]. The SRA method uses the midpoints like RMD, but adds a displacement of a suitable variance to all of the points to increase stability of the generated sequence [22].

Figure 4 shows how the SRA method generates an approximate self-similar sequence. The reason for interpolating midpoints is to construct Gaussian increments of X, which are correlated. Adding offsets to all points should make the resulted sequence self-similar and of normal distribution [22].

The SRA method consists of the following steps:

Step.1 If the process $\{X_t\}$ is to be computed for times instances t between 0 and 1, then start out by setting $X_0 = 0$ and selecting X_1 as a pseudo-random number from a Gaussian distribution with mean 0 and variance $Var[X_1] = \sigma_0^2$. Then $Var[X_1 - X_0] = \sigma_0^2$.

Step.2 Next, $X_{\frac{1}{2}}$ is constructed by the interpolation of the midpoint, that is, $X_{\frac{1}{2}} = \frac{1}{2}(X_0 + X_1)$.

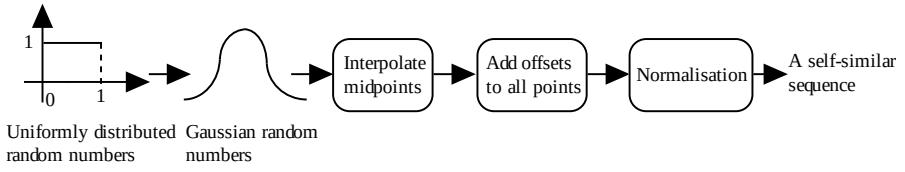

Figure 4: SRA method

Step.3 Add a displacement of a suitable variance to all of the points, i.e., $X_0 = X_0 + d_{1,1}, X_{\frac{1}{2}} = X_{\frac{1}{2}} + d_{1,2}, X_1 = X_1 + d_{1,3}$. The offsets $d_{1,*}$ are governed by fractional Gaussian noise. For $Var[X_{t_2} - X_{t_1}] = |t_2 - t_1|^{2H}\sigma_0^2$ to be true, for any $t_1, t_2, 0 \leq t_1 \leq t_2 \leq 1$, it is required that $Var[X_{\frac{1}{2}} - X_0] = \frac{1}{4}Var[X_1 - X_0] + 2S_1^2 = (\frac{1}{2})^{2H}\sigma_0^2$, that is, $S_1^2 = \frac{1}{2}(\frac{1}{2^1})^{2H}(1 - 2^{2H-2})\sigma_0^2$.

Step.4 Next, Step.2 and Step.3 are repeated. Therefore, $S_n^2 = \frac{1}{2}(\frac{1}{2^n})^{2H}(1 - 2^{2H-2})\sigma_0^2$, where σ_0^2 is an initial variance and $0 < H < 1$.

Using the above steps, the SRA method generates an approximate self-similar FBM process.

4 Analysis of Self-Similar Sequences

Three generators of self-similar sequences of pseudo-random numbers described in the Section 3 have been implemented in C on a Sun SPARCstation 4 (110 MHz, 32 MB), and used to generate self-similar cumulative arrival processes, mentioned at the end of Section 2. The mean times required for generating sequences of a given length were obtained by using the SunOS 5.5 `date` command and averaged over 30 iterations, having generated sequences of 32,768 (2^{15}), 131,072 (2^{17}), 262,144 (2^{18}), 524,288 (2^{19}) and 1,048,576 (2^{20}) numbers.

We have also analysed the efficiency of these methods in the sense of their accuracy. For each of $H = 0.5, 0.55, 0.7, 0.9, 0.95$, each method was used to generate over 100 sample sequences of 32,768 (2^{15}) numbers starting from different random seeds. Self-similarity and marginal distributions of the generated sequences were assessed by applying the best currently available techniques. These include:

- *Anderson-Darling goodness-of-fit test*: used to show that the marginal distribution of sample sequences generated by all three methods is normal or almost normal, since all three methods are based on Gaussian processes. This test is more powerful than *Kolmogorov-Smirnov* when testing against a specified normal distribution [7].

- *Sequence plot*: used to show that a generated sequence has LRD properties with the assumed H value.

- *Periodogram plot*: used to show whether a generated sequence is LRD or not. It can be shown that if the autocorrelations were summable, then near the origin the periodogram should be scattered randomly around a constant. If the autocorrelations were non-summable, i.e., LRD, the points of a sequence are scattered around a negative slope. The periodogram plot is obtained by plotting $log_{10}(periodogram)$ against $log_{10}(frequency)$. An estimate of the Hurst parameter is given by $\hat{H} = (1 - \hat{\beta}_3)/2$ where $\hat{\beta}_3$ is the slope [2].

- *R/S statistic plot*: graphical R/S analysis of empirical data can be used to estimate the Hurst parameter $\hat{H}$. An estimate of H is given by the asymptotic slope $\hat{\beta}_2$ of the R/S statistic plot, i.e., $\hat{H} = \hat{\beta}_2$ [2].

- *Variance-time plot*: is obtained by plotting $log_{10}(Var(X^{(m)}))$ against $log_{10}(m)$ and by fitting a simple least square line through the resulting points in the plane. An estimate of the Hurst parameter is given by $\hat{H} = 1 - \hat{\beta}_1/2$ where $\hat{\beta}_1$ is the slope [2].

- *Whittle's approximate maximum likelihood estimate(MLE)*: is a more refined data analysis method to obtain confidence intervals (CIs) for the Hurst parameter H [2].

4.1 Analysis of Accuracy

We have summarised the results of our analysis in the following:

- Anderson-Darling goodness-of-fit test was applied to test normality of sample sequences. The results of the tests, executed at the 5% significance level, showed that the generated sequences could be considered as normally distributed for all but a few sequences with the high value of H; as see Table 1 for Anderson-Darling test and Table 2 for Kolmogorov-Smirnov test, the former test is more powerful than the latter test when testing against a specified normal distribution.
- Sequence plots in Figure 5-7 show higher levels of correlation of data as the H value increases. In other words, generated sequences have the evidence of LRD properties.

The estimates of Hurst parameter obtained from the periodogram, the R/S statistic, the variance-time and Whittle's MLE, have been used to compare the accuracy of the three methods. The relative inaccuracy ΔH is calculated using the formula: $\Delta H = \frac{\hat{H}-H}{H} * 100\%$, where H is the input value and $\hat{H}$ is an empirical mean value. The presented numerical results are all averaged over 100 sequences.

- The periodogram plots have slopes decreasing as H increases and also see Figure 8-10. The negative slopes of all our plots for $H = 0.5, 0.55, 0.7, 0.9, 0.95$ were the evidence of self-similarity. A comparison of relative inaccuracy ΔH of the estimated Hurst parameters of three methods using periodogram plot is given in Table 3; also see Figure 17. We see that in the most cases parameter H of the FFT method was closer to the required value than in the case of the RMD and SRA methods, although the relative inaccuracy degrades with increasing H (but never exceeds 6%). The analysis of periodograms suggest that the FFT method always produces self-similar sequences with positively biased H, while sequences produced by two other methods are negatively biased.
- The plots of R/S statistic clearly confirmed the self-similar nature of the generated sequences and also see Figure 11-13. The relative inaccuracy ΔH of the estimated Hurst parameter, obtained by R/S statistic plot, is given in Table 4; also see Figure 18. As we see, these results suggest that the FFT method is slightly better than the other two (but for $H = 0.9, 0.95$). This method of analysis of H does not link any of these generators with persistently negative or positive bias of H, as the periodogram plots did.

Table 1: Percentages(%) of Anderson-Darling goodness-of-fit test for normality at the 5% significance level. Each size of sample sequences is 32,768 numbers.

	Theoretical Hurst parameter				
Method	0.5	0.55	0.7	0.9	0.95
FFT	100	100	98	59	34
RMD	97	97	98	64	38
SRA	97	97	95	58	32

Table 2: Percentages(%) of Kolmogorov-Smirnov goodness-of-fit test for normality at the 5% significance level. Each size of sample sequences is 32,768 numbers.

	Theoretical Hurst parameter				
Method	0.5	0.55	0.7	0.9	0.95
FFT	100	100	100	95	97
RMD	100	100	100	96	80
SRA	100	100	100	96	72

- The variance-time plots also supported the claim that generated sequences were self-similar and also see Figure 14-16. Table 5 gives the relative inaccuracy ΔH of the estimated Hurst parameters obtained by the variance-time plot; also see Figure 19. Again, all three methods show comparable quality of the output sequences in the sense of H, with the relative inaccuracy increasing with the increase in H, but remaining below 8%. This time, all results but one suggest that the output sequences are negatively biased H, regardless of the method.

- The results for Whittle estimator of H with the corresponding 95% CIs $\hat{H} \pm 1.96\hat{\sigma}_{\hat{H}}$, see Table 6, show that for all input H values, CIs for the FFT method cover the assumed theoretical values, while the RMD and SRA methods produce sequences weaker correlated than expected (except $H = 0.5$).

Table 3: Relative inaccuracy ΔH estimated from periodogram plots.

H	FFT	RMD	SRA
0.5	+ 0.07 %	- 0.01 %	- 0.09 %
0.55	+ 1.26 %	- 1.31 %	- 1.41 %
0.7	+ 3.14 %	- 3.74 %	- 3.78 %
0.9	+ 3.93 %	- 5.10 %	- 5.13 %
0.95	+ 3.99 %	- 5.28 %	- 5.31 %

Table 4: Relative inaccuracy ΔH estimated from R/S statistic plots.

H	FFT	RMD	SRA
0.5	+7.34 %	+8.74 %	+8.71 %
0.55	+5.32 %	+6.28 %	+6.23 %
0.7	+0.82 %	+1.28 %	+1.26 %
0.9	- 5.02 %	- 4.46 %	- 4.44 %
0.95	- 6.89 %	- 6.34 %	- 6.31 %

Table 5: Relative inaccuracy ΔH estimated from variance-time plots.

H	FFT	RMD	SRA
0.5	- 0.85 %	+0.57 %	- 2.76 %
0.55	- 1.00 %	- 0.19 %	- 2.97 %
0.7	- 1.88 %	- 1.76 %	- 3.38 %
0.9	- 5.39 %	- 5.29 %	- 6.00 %
0.95	- 6.98 %	- 6.91 %	- 7.47 %

Table 6: Estimated mean values of H using Whittle's MLE. Each CI is for over 100 sample sequences. 95% CIs for the means are given in parentheses.

Method	Theoretical Hurst parameter				
	.5	.55	.7	.9	.95
FFT	.500 (.490, .510)	.550 (.540, .560)	.700 (.691, .710)	.900 (.891, .909)	.949 (.940, .958)
RMD	.500 (.490, .510)	.538 (.528, .548)	.658 (.647, .666)	.826 (.817, .835)	.870 (.861, .879)
SRA	.500 (.490, .510)	.538 (.528, .547)	.656 (.647, .666)	.825 (.816, .834)	.869 (.860, .878)

Table 7: Complexity and mean running times of generators. Running times were obtained by using the SunOS 5.5 `date` command on a Sun SPARCstation 4 (110 MHz, 32 MB); each mean is averaged over 30 iterations.

Method	Complexity	Sequence of				
		32,768 Numbers	131,072 Numbers	262,144 Numbers	524,288 Numbers	1,048,576 Numbers
		Mean running time (*minute:second*)				
FFT	$O(n\log n)$	0:5	0:20	0:35	1:12	3:47
RMD	$O(n)$	0:3	0:11	0:29	0:40	1:33
SRA	$O(n)$	0:3	0:10	0:20	0:40	1:31

Our results show that all three generators produce approximately self-similar sequences, with the relative inaccuracy ΔH increasing with H, but always staying below 10%. Apparently there is a problem with more detailed comparative studies of such generators, since different methods of analysis of the Hurst parameter can give very different results regarding the bias of $\hat{H}$ characterising the same output sequences. More reliable methods for assessment of self-similarity in pseudo-random sequences are needed.

4.2 Computational Complexity

The results of our experimental analysis of mean times needed by the three generators for generating pseudo-random self-similar sequences of a given length are shown in Table 7. The main conclusions are listed below.

- *FFT method* is the slowest of the three analysed methods. This is caused by relatively high complexity of the inverse FFT algorithm. Table 7 shows its time complexity and the mean running time. It took 5 seconds to generate a sequence of 32,768 (2^{15}) numbers, while generation of a sequence with 1,048,576 (2^{20}) numbers took 3 minutes and 47 seconds. FFT method requires $O(n\log n)$ computations to generate n numbers [19], [23].
- *RMD method* is faster and simpler than FFT. Table 7 shows its time complexity and the mean running time. Generation of a sequence with 32,768 (2^{15}) numbers took 3 seconds. It also took 1 minute and 33 seconds to generate a sequence of 1,048,576 (2^{20}) numbers. The theoretical algorithmic complexity is $O(n)$ [22].
- *SRA method* appears to be as fast as RMD. Table 7 shows its time complexity and the mean running time. The theoretical algorithmic complexity is $O(n)$ [22].

In summary, our results show that the generators based on RMD and SRA are faster in practical applications than the generator based on FFT, when long self-similar sequences of numbers are needed.

5 Conclusions

In this paper we have presented the results of a comparative analysis of three generators of (long) pseudo-random self-similar sequences. It appears that all three generators, based on FFT, RMD and

SRA, generate approximately self-similar sequences, with the relative inaccuracy of the resulted H below 9%, if $0.5 \leq H \leq 0.95$. On the other hand, the analysis of mean times needed for generating sequences of given lengths shows that two generators (based on RMD and SRA) should be recommended for practical simulation of telecommunication networks, since they are much faster than the generator based on FFT. Our study has also revealed that a robust method for comparative studies of self-similarity in pseudo-random sequences is needed, since currently available methods can provide inconclusive proofs of accuracy of such sequences. This is the direction of our current research.

References

[1] Beran, J. Statistical Methods for Data with Long Range Dependence. *Statistical Science 7*, 4 (1992), 404–427.

[2] Beran, J. *Statistics for Long-Memory Processes*. Chapman and Hall, An International Thomson Publishing Company, 1994.

[3] Cario, M., and Nelson, B. Numerical Methods for Fitting and Simulating Autoregressive-to-Anything Processes. *INFORMS Journal on Computing 10*, 1 (1998), 72–81.

[4] Cox, D. Long-Range Dependence: a Review. In *Statistics: An Appraisal* (1984), H. David and H. David, Eds., Iowa State Statistical Library, The Iowa State University Press, pp. 55–74.

[5] Crilly, A., Earnshaw, R., and Jones, H. *Fractals and Chaos*. Springer-Verlag, 1991.

[6] Garrett, M., and Willinger, W. Analysis, Modeling and Generation of Self-Similar VBR Video Traffic. In *Computer Communication Review Proceedings of ACM SIGCOMM'94* (London, UK, Aug. 1994), vol. 24(4), pp. 269–280.

[7] Gibbons, J., and Chakraborti, S. *Nonparametric Statistical Inference*. Marcel Dekker, Inc., 1992.

[8] Granger, C. Long Memory Relationships and the Aggregation of Dynamic Models. *Journal of Econometrics 14* (1980), 227–238.

[9] Hosking, J. Modeling Persistence in Hydrological Time Series Using Fractional Differencing. *Water Resources Research 20*, 12 (Dec. 1984), 1898–1908.

[10] Jeong, H.-D., McNickle, D., and Pawlikowski, K. A Generator of Pseudo-random Self-Similar Sequences Based on SRA. Tech. Rep. TR-COSC 10/98, Department of Computer Science, University of Canterbury, Christchurch, New Zealand, 1998.

[11] Krunz, M., and Makowski, A. A Source Model for VBR Video Traffic Based on $M/G/\infty$ Input Processes. In *Proceedings of IEEE INFOCOM'98* (San Francisco, CA, USA, Mar. 1998), pp. 1441–1448.

[12] Lau, W.-C., Erramilli, A., Wang, J., and Willinger, W. Self-Similar Traffic Generation: the Random Midpoint Displacement Algorithm and Its Properties. In *Proceedings of IEEE ICC'95* (Seattle, WA, 1995), pp. 466–472.

[13] Leland, W., Taqqu, M., Willinger, W., and Wilson, D. On the Self-Similar Nature of Ethernet Traffic(Extended Version). *IEEE/ACM Transactions on Networking 2*, 1 (Feb. 1994), 1–15.

[14] Likhanov, N., Tsybakov, B., and Georganas, N. Analysis of an ATM Buffer with Self-Similar("Fractal") Input Traffic. In *Proceedings of IEEE INFOCOM'95* (1995), pp. 985–992.

[15] Mandelbrot, B. A Fast Fractional Gaussian Noise Generator. *Water Resources Research 7* (1971), 543–553.

[16] Mandelbrot, B., and Wallis, J. Computer Experiments with Fractional Gaussian Noises. *Water Resources Research 5*, 1 (1969), 228–267.

[17] Neidhardt, A., and Wang, J. The Concept of Relevant Time Scales and Its Application to Queueing Analysis of Self-Similar Traffic (or Is Hurst Naughty or Nice?). In *Proceedings ACM SIGMETRICS'98* (Madison, Wisconsin, USA, Jun. 1998), pp. 222–232.

[18] Norros, I. A Storage Model with Self-Similar Input. *Queueing Systems 16* (1994), 387–396.

[19] Paxson, V. Fast Approximation of Self-Similar Network Traffic. Tech. Rep. LBL-36750, Lawrence Berkeley Laboratory and EECS Division,University of California, Berkeley, Apr. 1995.

[20] Paxson, V., and Floyd, S. Wide-Area Traffic: the Failure of Poisson Modeling. *IEEE/ACM Transactions on Networking 3*, 3 (Jun. 1995), 226–244.

[21] Peitgen, H.-O., Jurgens, H., and Saupe, D. *Chaos and Fractals: New Frontiers of Science.* Springer-Verlag, 1992.

[22] Peitgen, H.-O., and Saupe, D. *The Science of Fractal Images.* Springer-Verlag, 1988.

[23] Press, W., Flannery, B., Teukolsky, S., and Vetterling, W. *Numerical Recipes: The Art of Scientific Computing.* Cambridge University Press, 1986.

[24] Rose, O. *Traffic Modeling of Variable Bit Rate MPEG Video and Its Impacts on ATM Networks.* PhD thesis, Bayerische Julus-Maximilians-Universitat Wurzburg, 1997.

[25] Ryu, B. *Fractal Network Traffic: from Understanding to Implications.* PhD thesis, Graduate School of Arts and Science, Columbia University, 1996.

[26] Taralp, T., Devetsikiotis, M., and Lambadaris, I. Efficient Fractional Gaussian Noise Generation Using the Spatial Renewal Process. In *Proceedings IEEE International Communications Conference (ICC'98)* (Atlanta, GA, USA, Jun. 1998), pp. S41–3.1–S41–3.5.

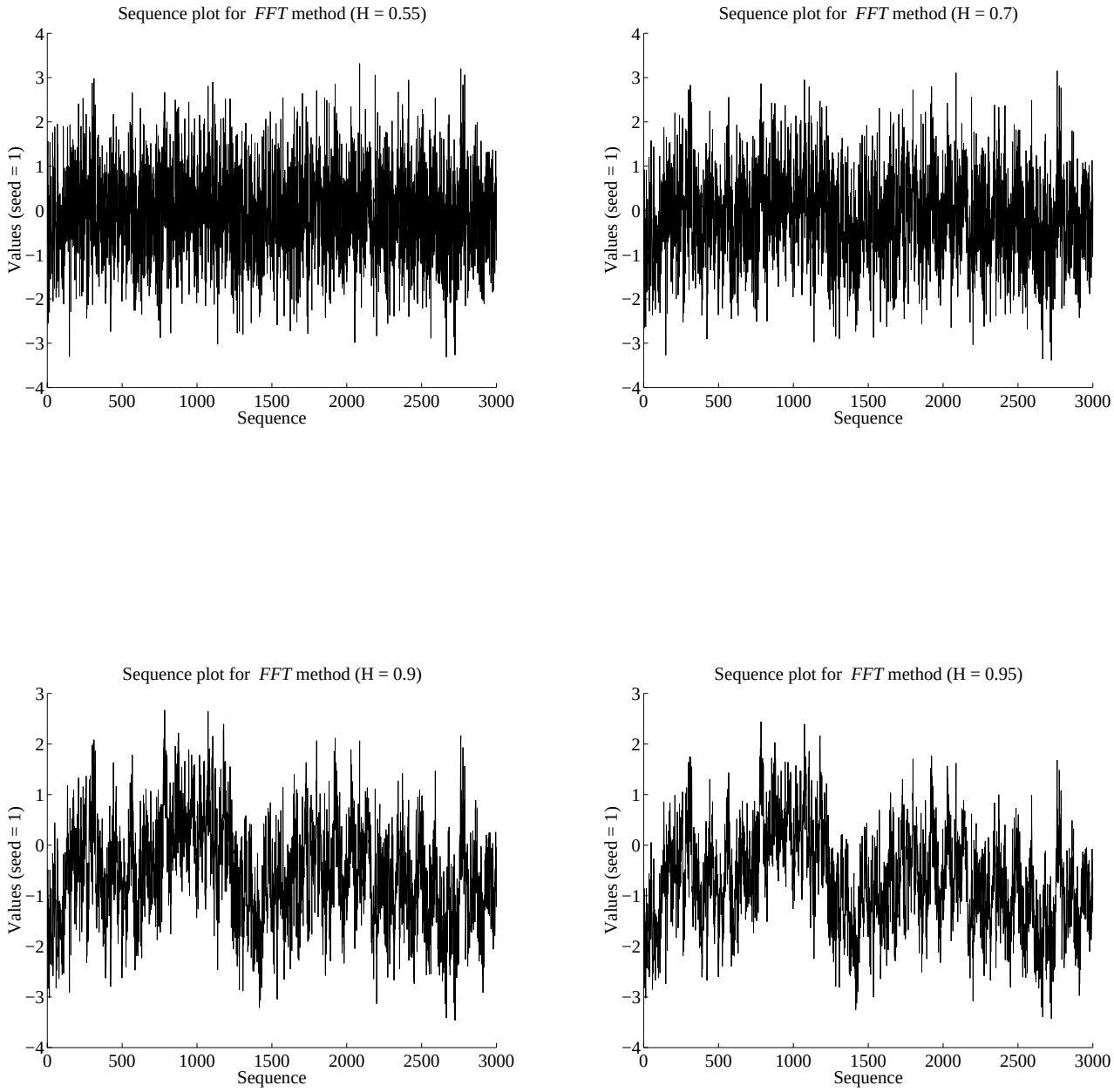

Figure 5: Sequence plots for FFT method (H = 0.55, 0.7, 0.9, 0.95).

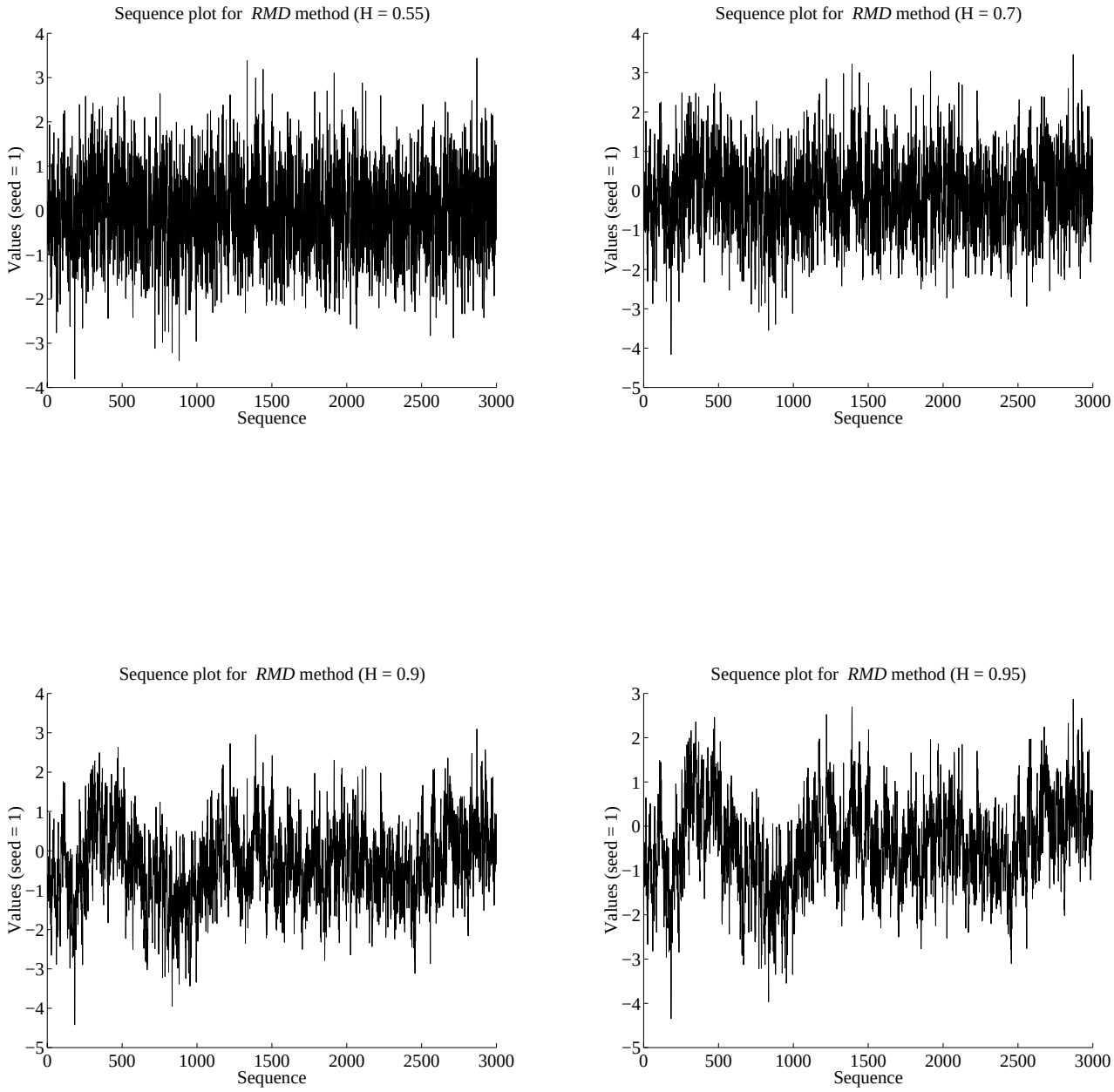

Figure 6: Sequence plots for RMD method (H = 0.55, 0.7, 0.9, 0.95).

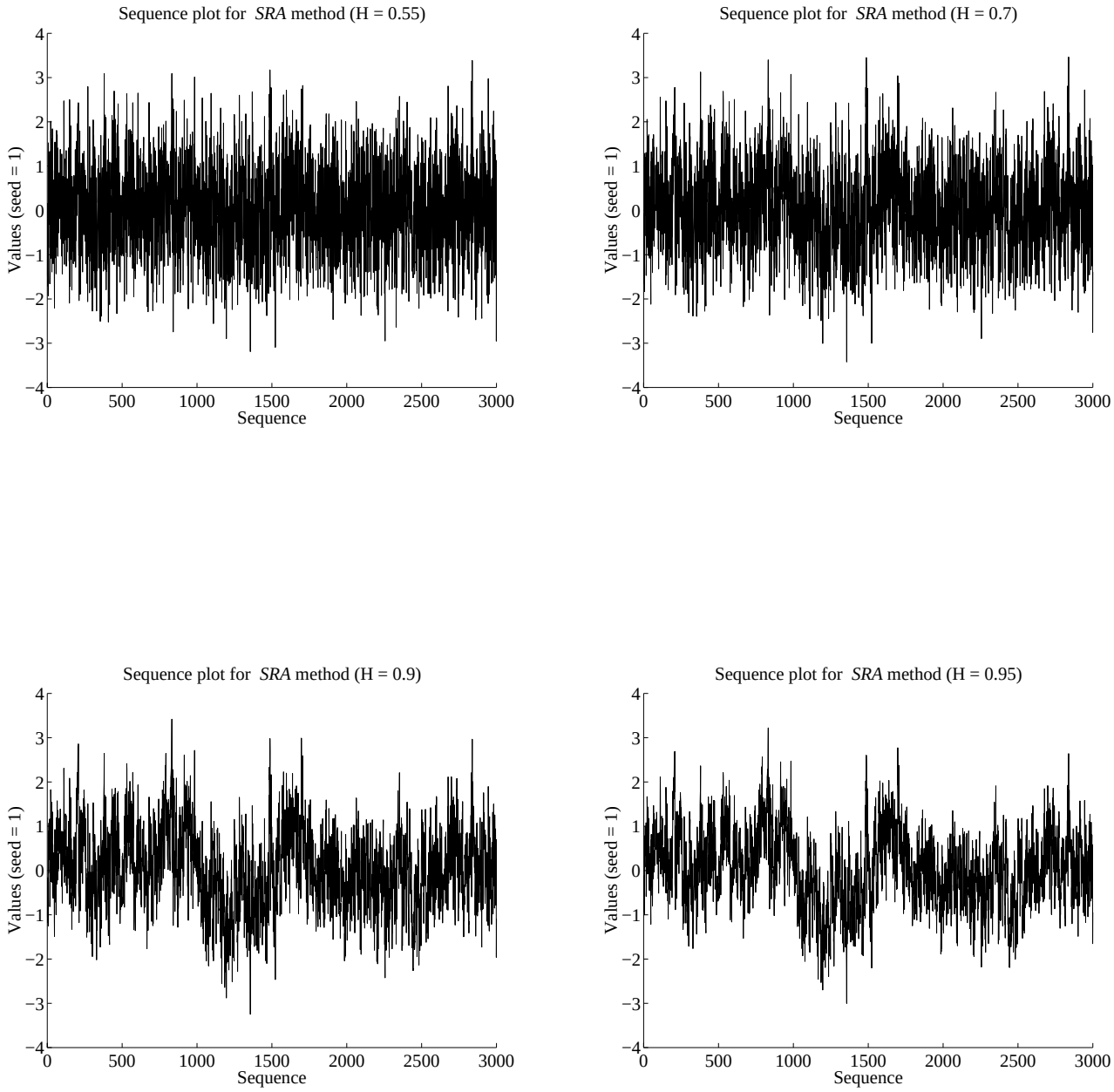

Figure 7: Sequence plots for SRA method (H = 0.55, 0.7, 0.9, 0.95).

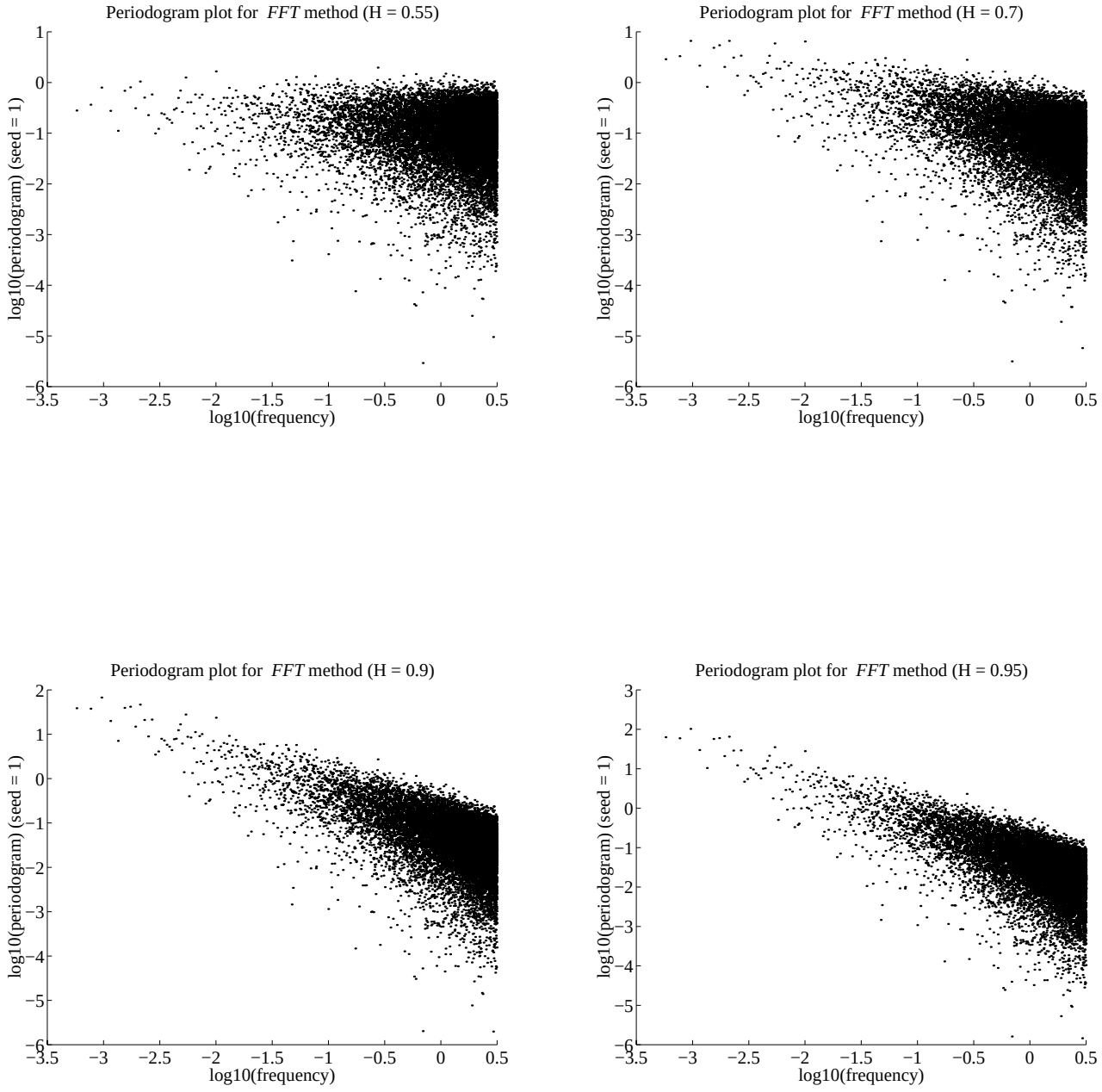

Figure 8: Periodogram plots for FFT method (H = 0.55, 0.7, 0.9, 0.95).

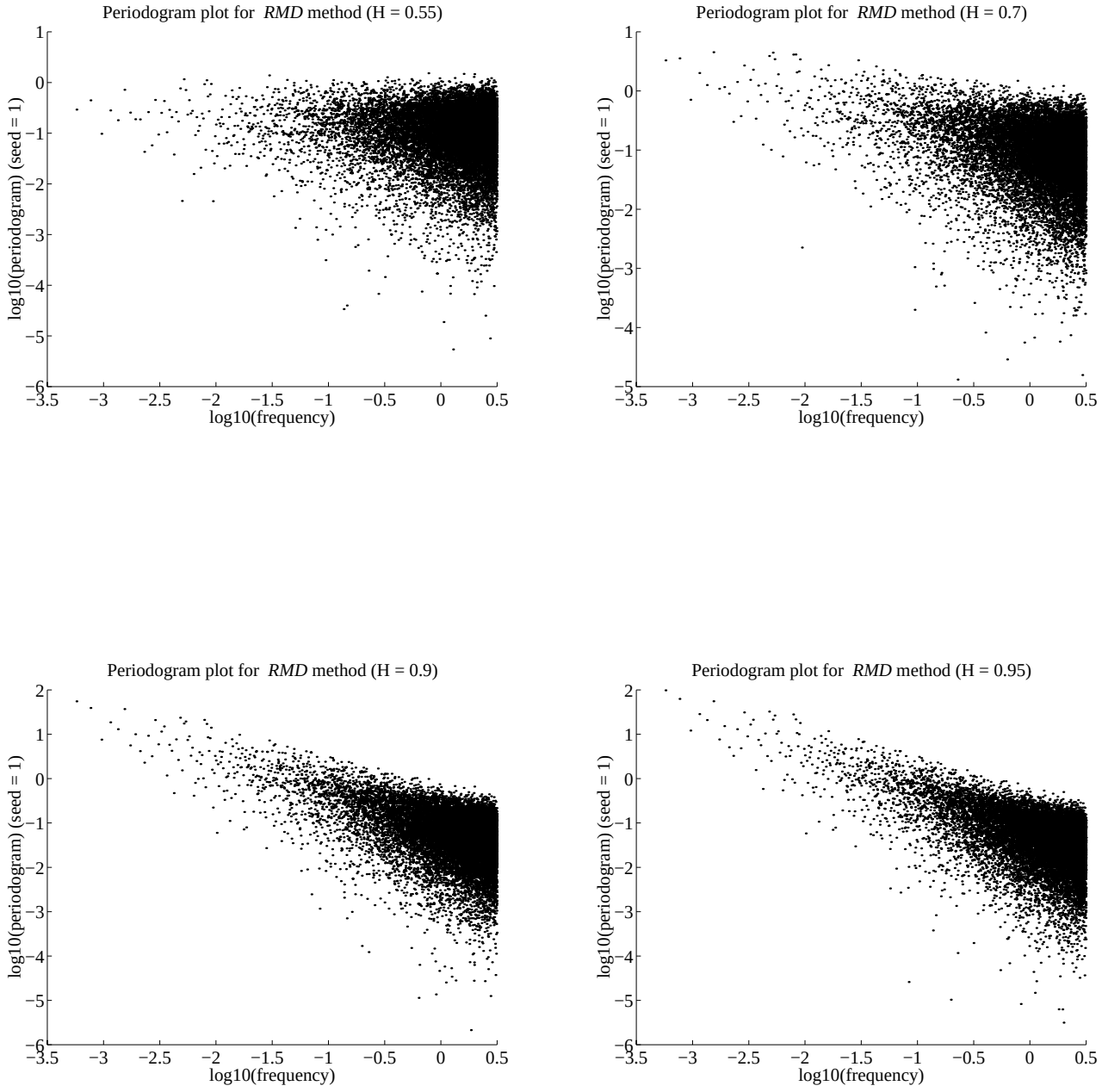

Figure 9: Periodogram plots for RMD method (H = 0.55, 0.7, 0.9, 0.95).

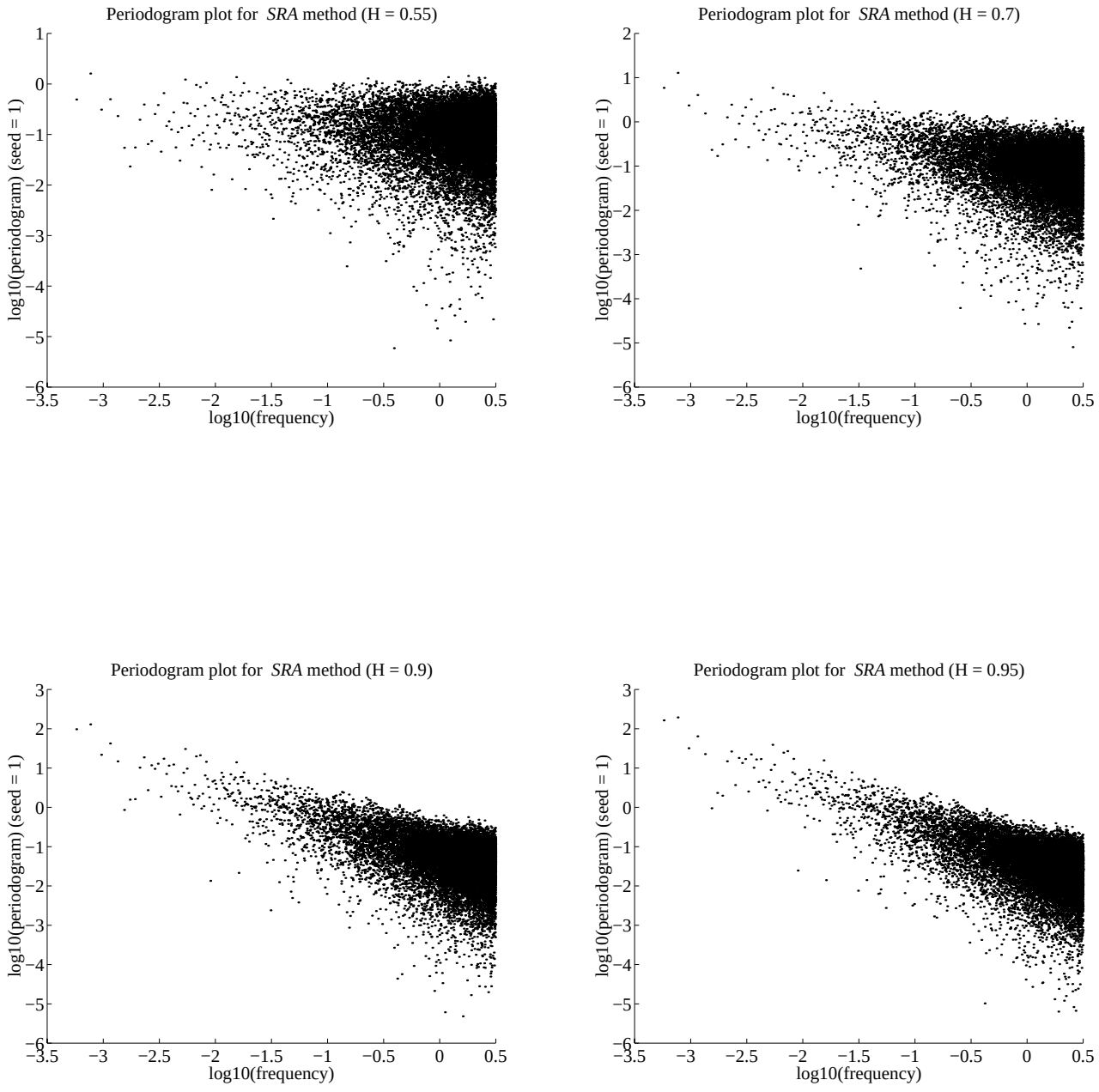

Figure 10: Periodogram plots for SRA method (H = 0.55, 0.7, 0.9, 0.95).

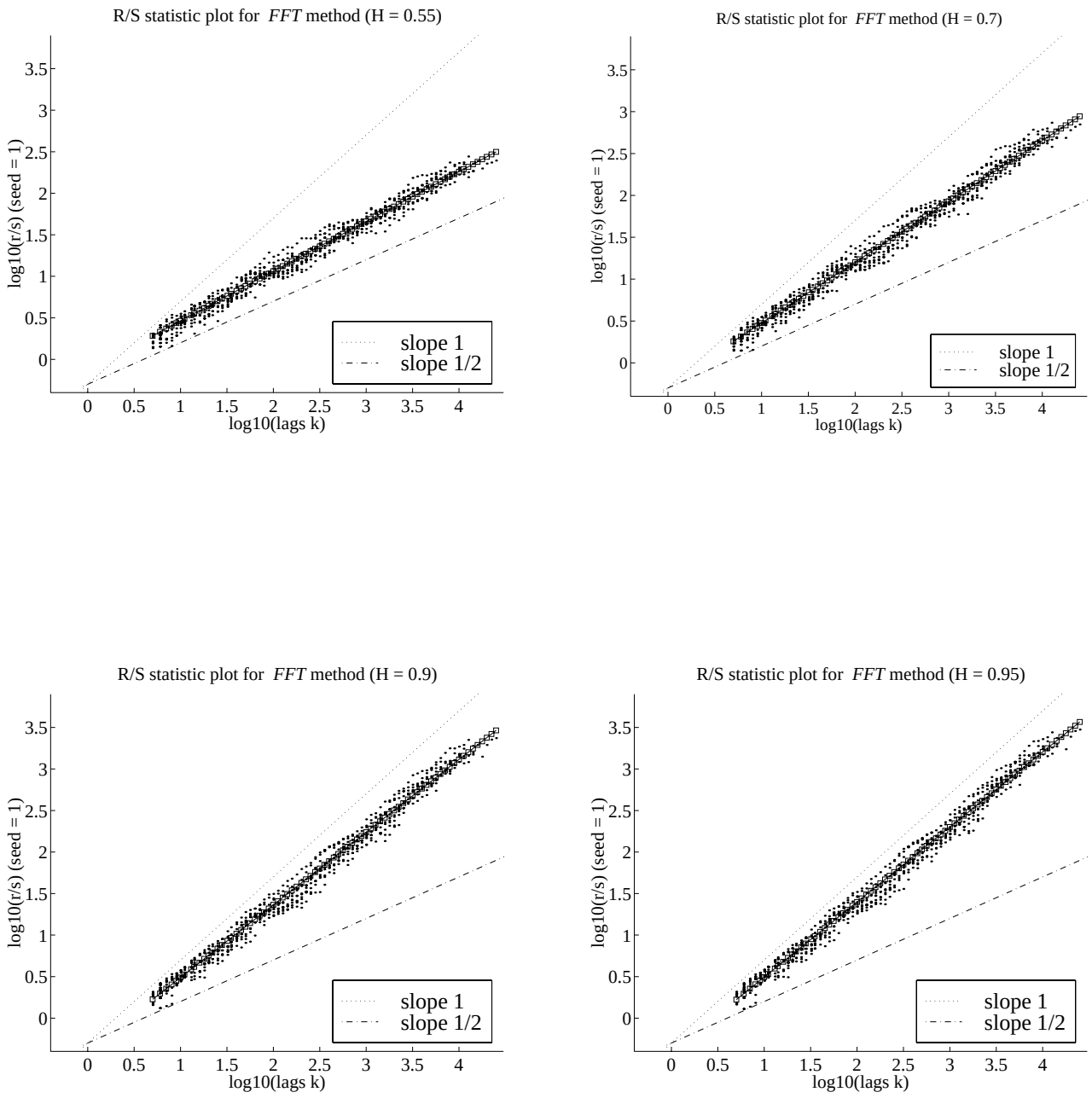

Figure 11: R/S statistic plots for FFT method (H = 0.55, 0.7, 0.9, 0.95).

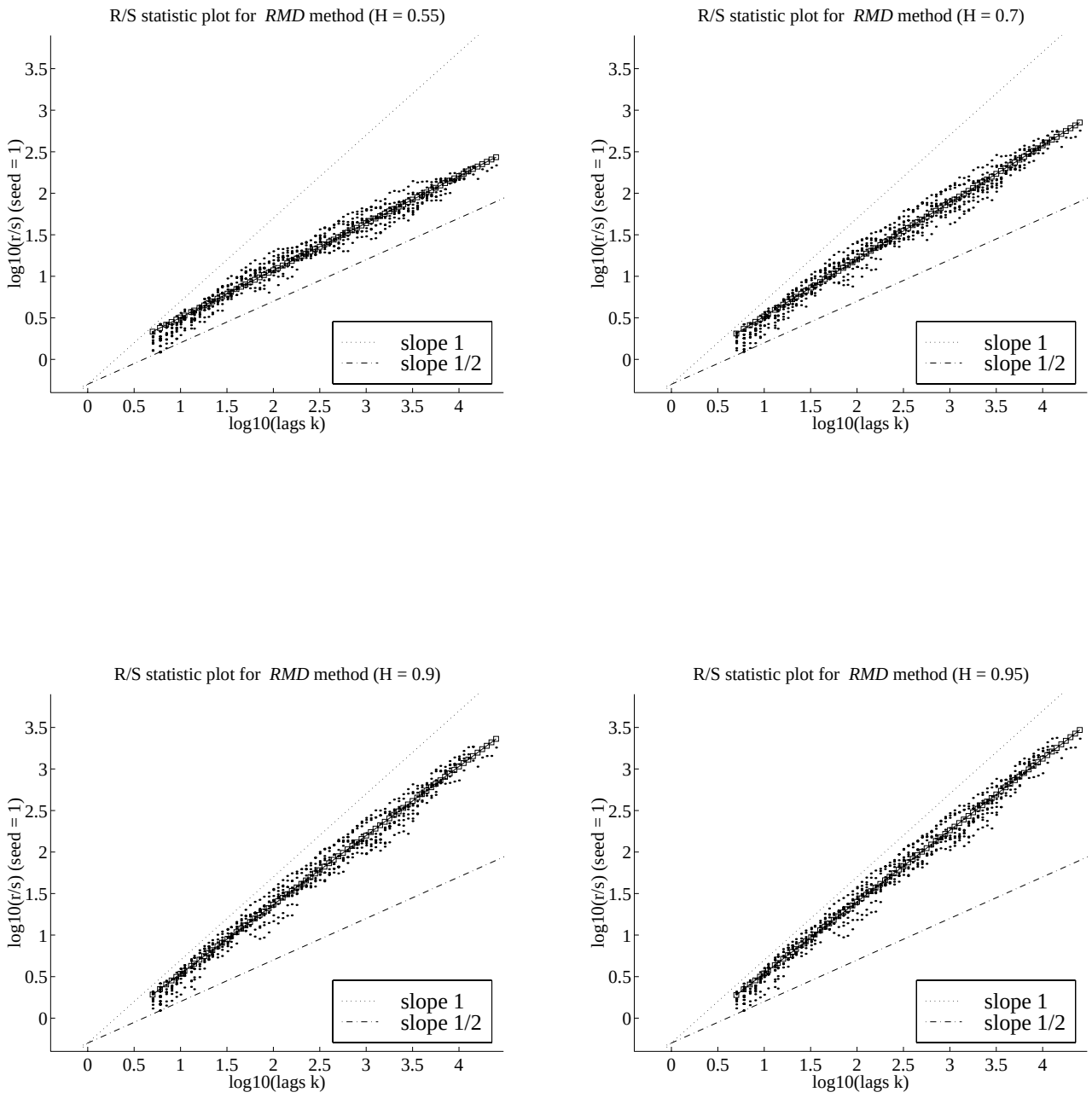

Figure 12: R/S statistic plots for RMD method (H = 0.55, 0.7, 0.9, 0.95).

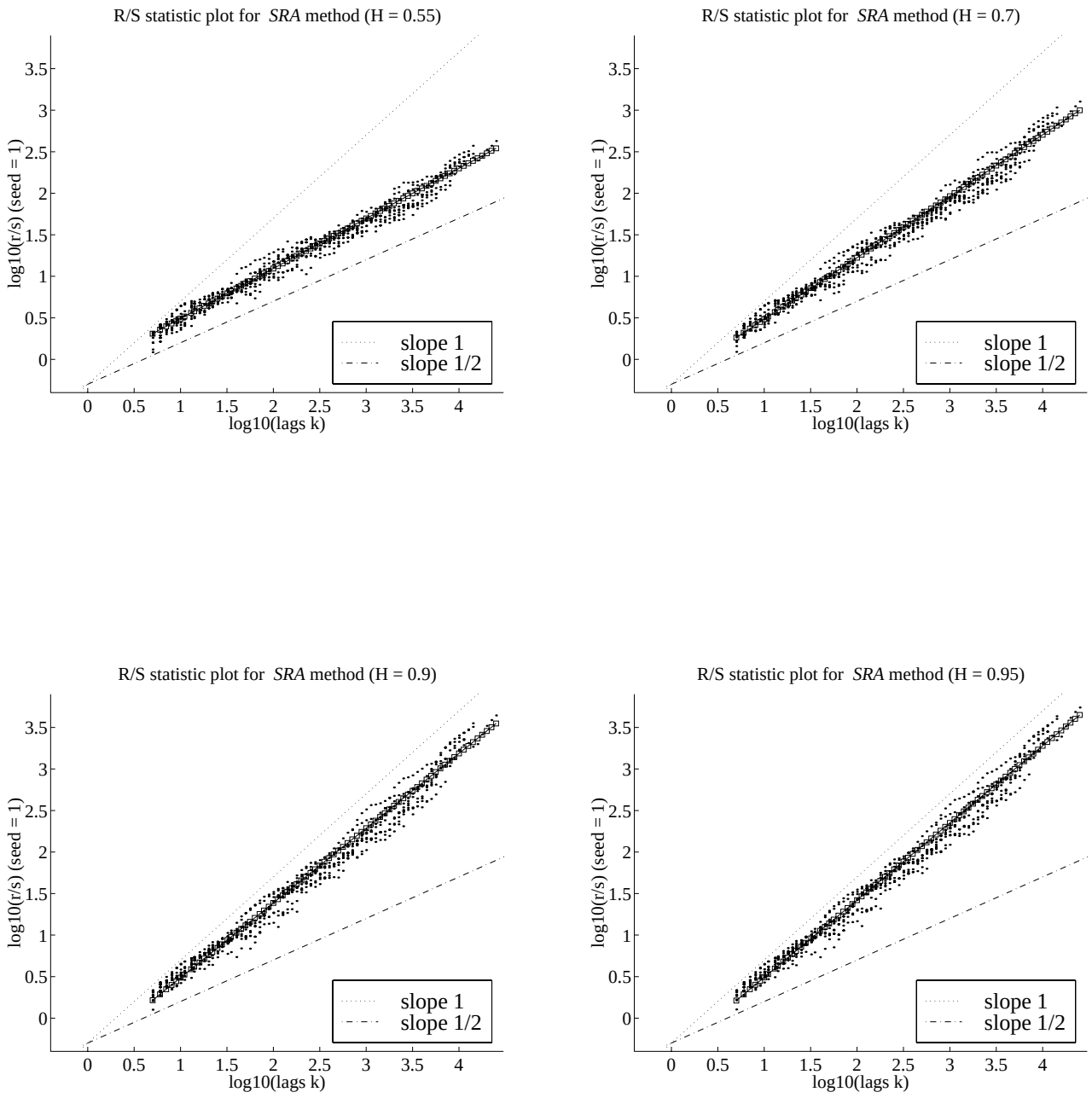

Figure 13: R/S statistic plots for SRA method (H = 0.55, 0.7, 0.9, 0.95).

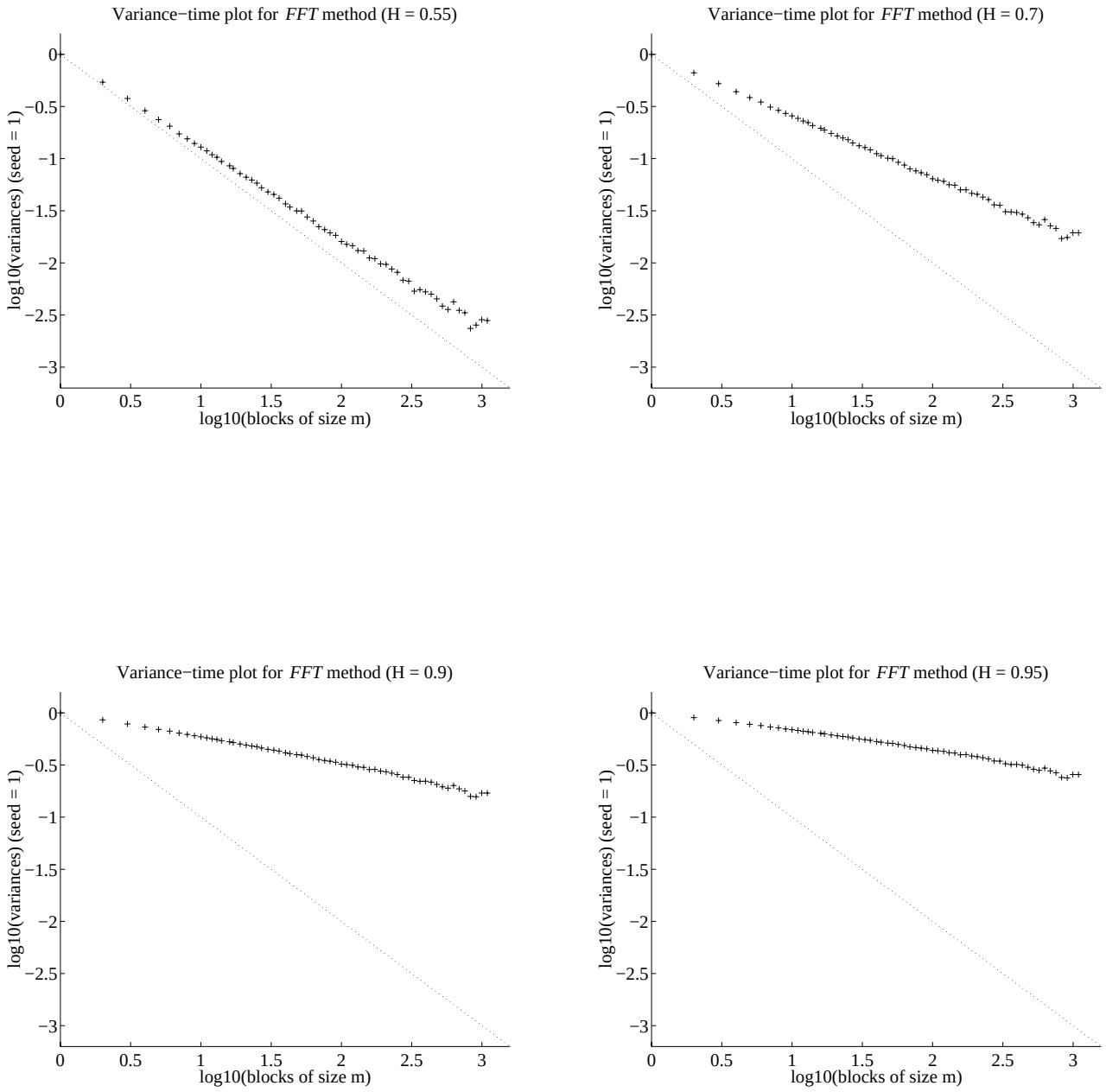

Figure 14: Variance-time plots for FFT method (H = 0.55, 0.7, 0.9, 0.95).

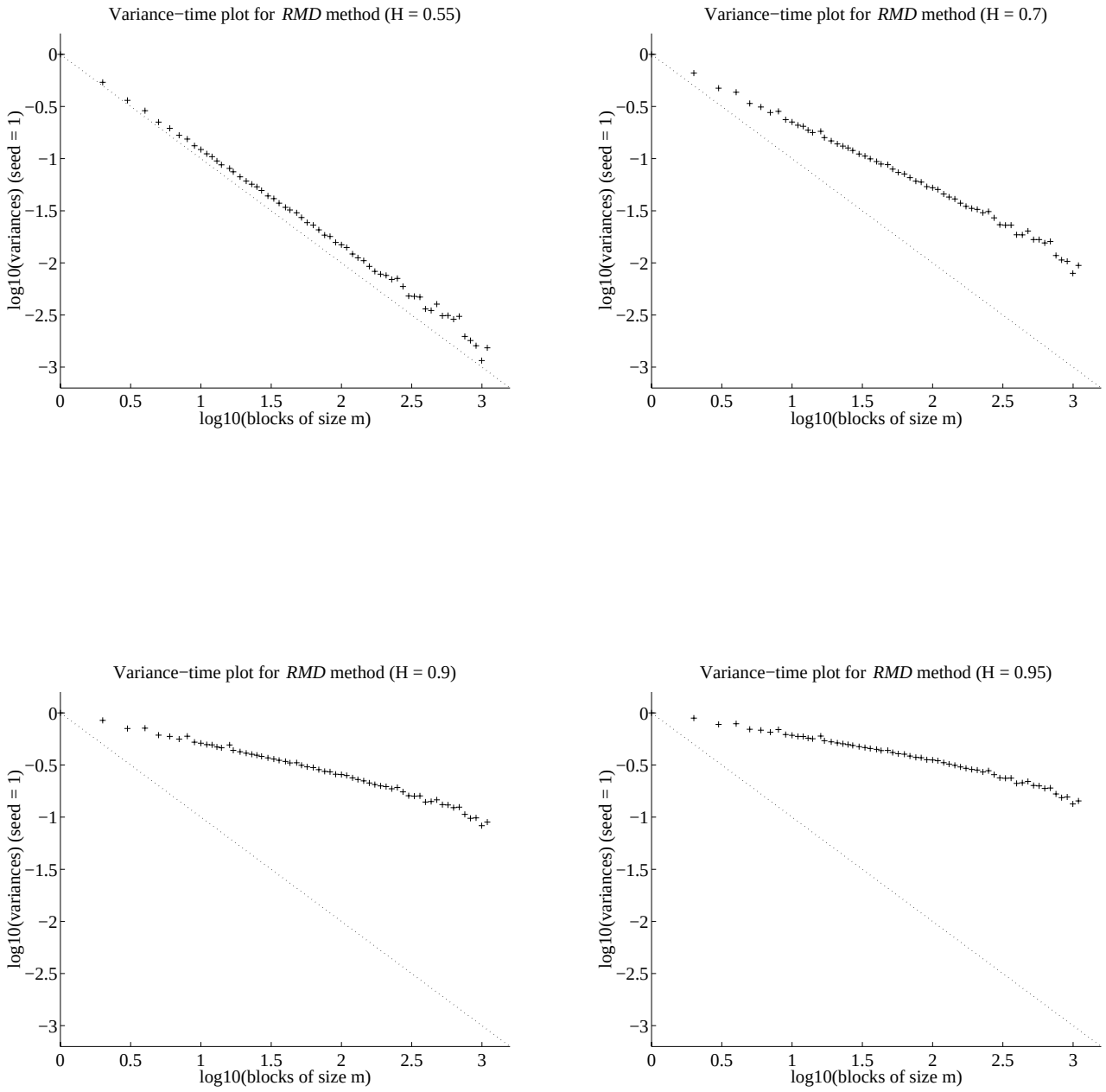

Figure 15: Variance-time plots for RMD method (H = 0.55, 0.7, 0.9, 0.95).

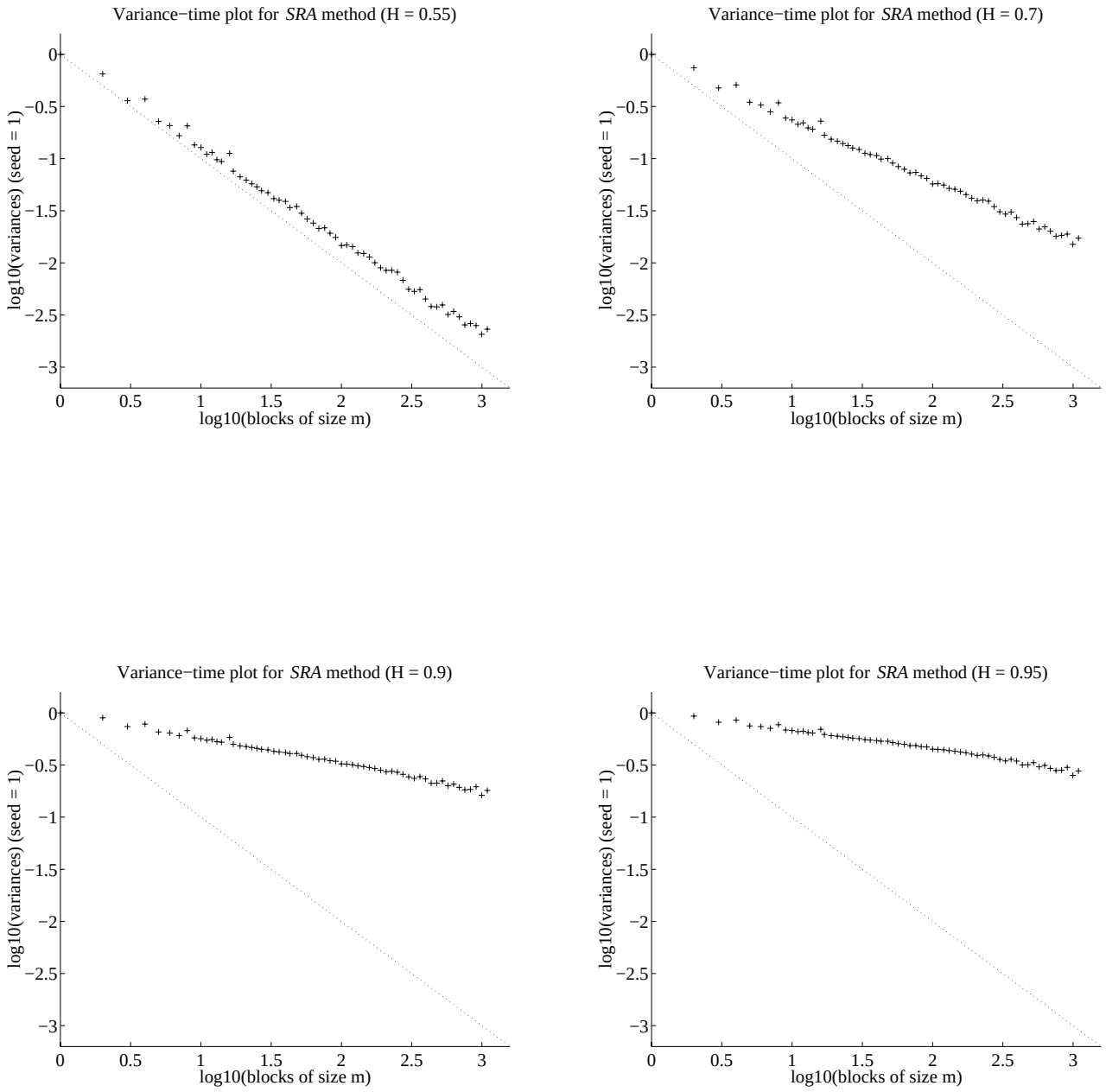

Figure 16: Variance-time plots for SRA method (H = 0.55, 0.7, 0.9, 0.95).

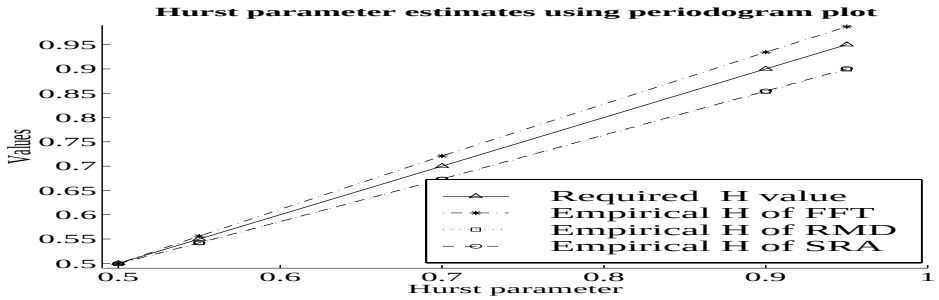

Figure 17: Estimation of Hurst parameter by periodogram plot

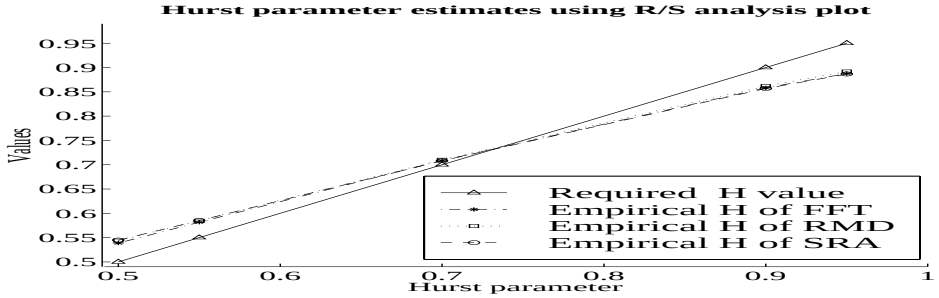

Figure 18: Estimation of Hurst parameter by R/S statistic plot

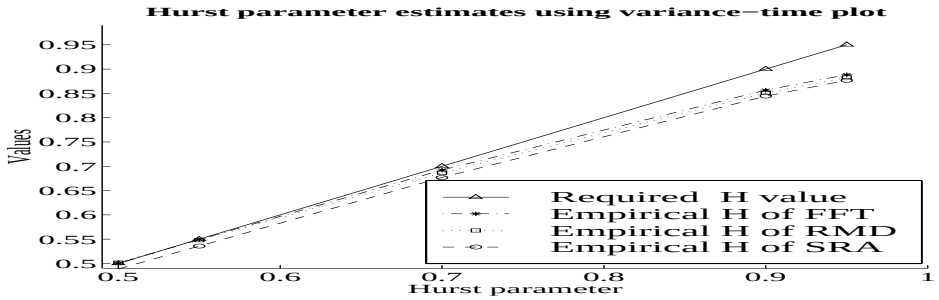

Figure 19: Estimation of Hurst parameter by variance-time plot